Supplementary Material: PuppeteerGAN: Arbitrary Portrait Animation with Semantic-aware Appearance Transformation

Zhuo Chen[1,2], Chaoyue Wang [2], Yuan Bo[1], and Dacheng Tao[2]

[1]Shenzhen International Graduate School, Tsinghua University
[2]UBTECH Sydney AI Centre, School of Computer Science, Faculty of Engineering, The University of Sydney, Darlington, NSW 2008, Australia
z-chen17@mails.tsinghua.edu.cn, chaoyue.wang@sydney.edu.au, yuanb@sz.tsinghua.edu.cn, dacheng.tao@sydney.edu.au

A. Network Architectures

PuppeteerGAN animates portraits in a two-stage pipeline including two networks: Sketching Network and Coloring Network. Sketching Network performs pose retargeting on the segmentation mask of the source portrait and the landmark of the target. The Coloring Network transforms the appearance from the generated segmentation mask to realistic portrait. In order to make full use of the texture information extracted by the shallow layers of encoder, we use the proposed warp-based semantic-aware skip-connections in the Coloring Network. The detailed network architectures are listed in Table 1 and Table 2.

B. Additional Comparisons and Ablation Studies

In this section, we demonstrate the effectiveness of the proposed PuppeteerGAN and the warp-based semantic-aware skip-connection by comparing the qualitative results of portrait animation. In the experiment, we compare PuppeteerGAN with two other face animation methods Averbuch-Elor *et al*. [1] and X2Face [6]. Considering our method is conditioned on the segmentation mask, two recently conditional image synthesis methods Pix2PixHD [5] and SPADE [4] are also used for comparison. In order to illustrate the effect of our warp-based semantic-aware skip-connection, we further report the results of the network with original skip-connections (*i.e*., Original skip-connection).

The test video sequences are selected from VoxCeleb [3], among which all the identities are unseen for PuppeteerGAN during training. For each sequence, we randomly select one frame as the source portrait and the others as the driven images.

In Fig. 1 and Fig. 2, we show two examples of portrait animation. The first row displays the source portrait and target frames. The results of Pix2PixHD [5] are reported in the second row, which demonstrate fidelity and semantically reasonable mimic face. However, it fails to recover the detailed texture of the source image. Then, although two warp based methods, Averbuch-Elor et al. [1] and X2Face [6], perform well when pose changes slightly, a large motion will cause significant artifacts. In addition, since the spatially-adaptive normalization (SPADE) [4] is employed in our Coloring network, we compare our method with the original SPADE [4]. As shown in the fifth row, SPADE is able to generate highly realistic portraits, but fails to preserve the texture (*i.e*. identity) of the source person. Furthermore, we replace the warp-based semantic-aware skip-connection with the simple skip-connection (the next-to-last row). Due to the lack of semantic guidance, the features from skip-connection may be geometry misaligned, which leads to global blur and detail artifacts. Finally, the last line shows the portraits animated by our PuppeteerGAN, which is superior to the existing and the ablation methods in both reality and fidelity.

C. Additional Qualitative Results

We provide more results of portrait animation in cross-identity/domain/resolution cases to demonstrate the fidelity, generalization and extensibility of PuppeteerGAN. Fig. 3 displays the cross-identity portrait animation results. Fig. 4, Fig. 5 and

Sketching Network						
Landmark Encoder($\mathbf{E}_L$)						
layer	**k**	**s**	**output**	**res**	**input**	**activation**
elconv1	3	2	64	2	landmark heatmap	ReLU
elconv2	3	2	64	4	elconv1	ReLU
elconv3	3	2	128	8	elconv2	ReLU
elconv4	3	2	128	16	elconv3	ReLU
elconv5	3	2	256	32	elconv4	ReLU
elconv6	3	2	256	64	elconv5	ReLU
elconv7	3	2	256	128	elconv6	ReLU
Segmentation Encoder($\mathbf{E}_M$)						
layer	**k**	**s**	**output**	**res**	**input**	**activation**
emconv0	9	1	32	1	segmentation mask	ReLU
emconv1	3	2	64	2	emconv0	ReLU
emconv2	3	2	64	4	emconv1	ReLU
emconv3	3	2	128	8	emconv2	ReLU
emconv4	3	2	128	16	emconv3	ReLU
emconv5	3	2	256	32	emconv4	ReLU
emconv6	3	2	256	64	emconv5	ReLU
emconv7	3	2	256	128	emconv6	ReLU
Sketching Generator($\mathbf{G}_I$)						
layer	**k**	**s**	**output**	**res**	**input**	**activation**
grdeconv1	3	2	256	64	elconv7$\oplus$ elconv6	BN+ReLU
grdeconv2	3	2	256	32	grdeconv1	BN+ReLU
grdeconv3	3	2	256	16	grdeconv2	BN+ReLU
grdeconv4	3	2	256	8	grdeconv3	BN+ReLU
grdeconv5	3	2	256	4	grdeconv4	BN+ReLU
grdeconv6	3	2	256	2	grdeconv5	BN+ReLU
grdeconv7	3	2	256	1	grdeconv6	BN+ReLU
grdeconv8	3	1	128	1	grdeconv7	BN+ReLU
grdeconv9	3	1	68+19	1	grdeconv8	Sigmoid
Sketching Discriminator($\mathbf{D}_I$)						
layer	**k**	**s**	**output**	**res**	**input**	**activation**
diconv1	3	1	256	128	elconv7	ReLU
diconv2	3	2	256	256	diconv1	ReLU
difc1	-	-	512	256	diconv2(pair)	-
difc2	-	-	256	256	difc1	-
difc3	-	-	1	256	difc2	-

Table 1: The architecture of Sketching Network.Here $\mathbf{k}$ and $\mathbf{s}$ represent the kernel size and the stride. $\mathbf{res}$ means the downscaling factor with regard to the input image size. $\mathbf{output}$ is the number of output channels, while input is the input items of each layer. The Sketching Network consists of two encoders (landmark and segmentation mask), a generator and a discriminator.

Fig. 6 show that PuppeteerGAN can animate portraits of diverse domains including color photos, black-and-white photos, paintings, sculptures and cartoon characters. We illustrate the results of animating high-resolution portraits in Fig. 7.

Coloring Network						
layer	**k**	**s**	**output**	**res**	**input**	**activation**
gconv1	7	1	64	1	image	LeakReLU(0.2)
gres2	3	2	64	2	gconv1	LeakReLU(0.2)
gres3	3	2	64	4	gres2	LeakReLU(0.2)
gres4	3	2	128	8	gres3	LeakReLU(0.2)
gres5	3	2	256	16	gres4	LeakReLU(0.2)
gres6	3	2	256	32	gres5	LeakReLU(0.2)
gres7	3	2	512	64	gres6	LeakReLU(0.2)
gres8	3	2	512	128	gres7	LeakReLU(0.2)
gderes7	3	2	512	64	gres8+gres7	BN+LeakReLU(0.2)
gderes6	3	2	256	32	gderes7+gres6	BN+LeakReLU(0.2)
gderes5	3	2	256	16	gderes6+gres5	BN+LeakReLU(0.2)
gderes4	3	2	128	8	gderes5+gres4	BN+LeakReLU(0.2)
gderes3	3	2	64	4	gderes4+gres3	BN+LeakReLU(0.2)
gderes2	3	2	64	2	gderes3+gres2	BN+LeakReLU(0.2)
gderes1	3	2	64	1	gderes2	BN+LeakReLU(0.2)
gconv1	3	1	64	1	gderes1	LeakReLU(0.2)
gconv0	3	1	3	1	gconv1	Tanh

Table 2: The architecture of Coloring Network. The layer gres and gderes means the residual block with down-sample and up-sample, respectively. The proposed warp-based semantic-aware skip-connection is shown as the plus of two inputs in this table.

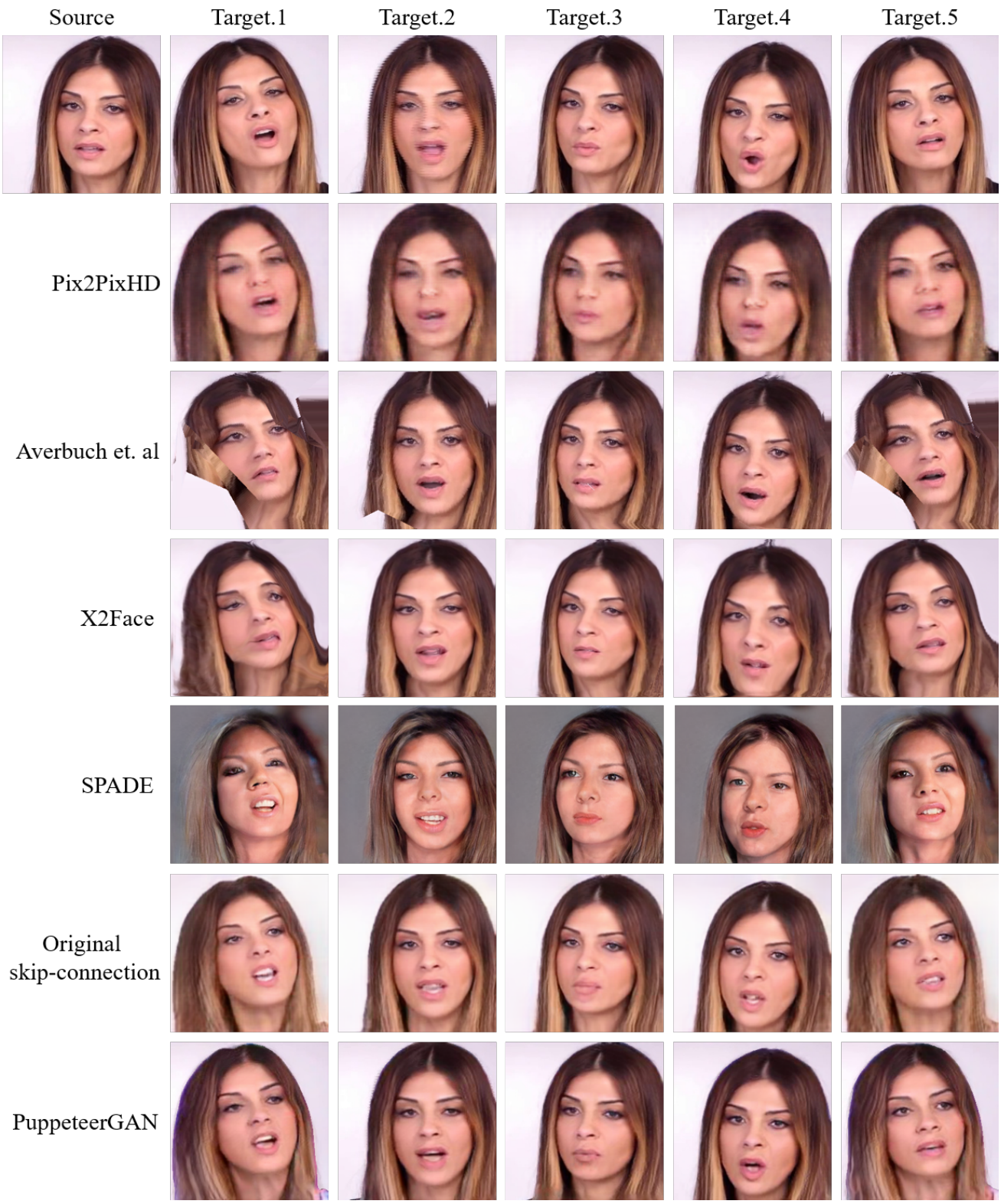

Figure 1: The results of a source portrait animated by the other frames in the same video.The first row shows the source portrait and five target frames. The source frame is selected randomly from the sequence in this experiment. We compare PuppeteerGAN with Pix2PixHD [5], Averbuch-Elor et al. [1], X2face [6], SPADE [4] and network with original skip-connections. For each method, only the source portrait can be used for fine-tuning training if needed.

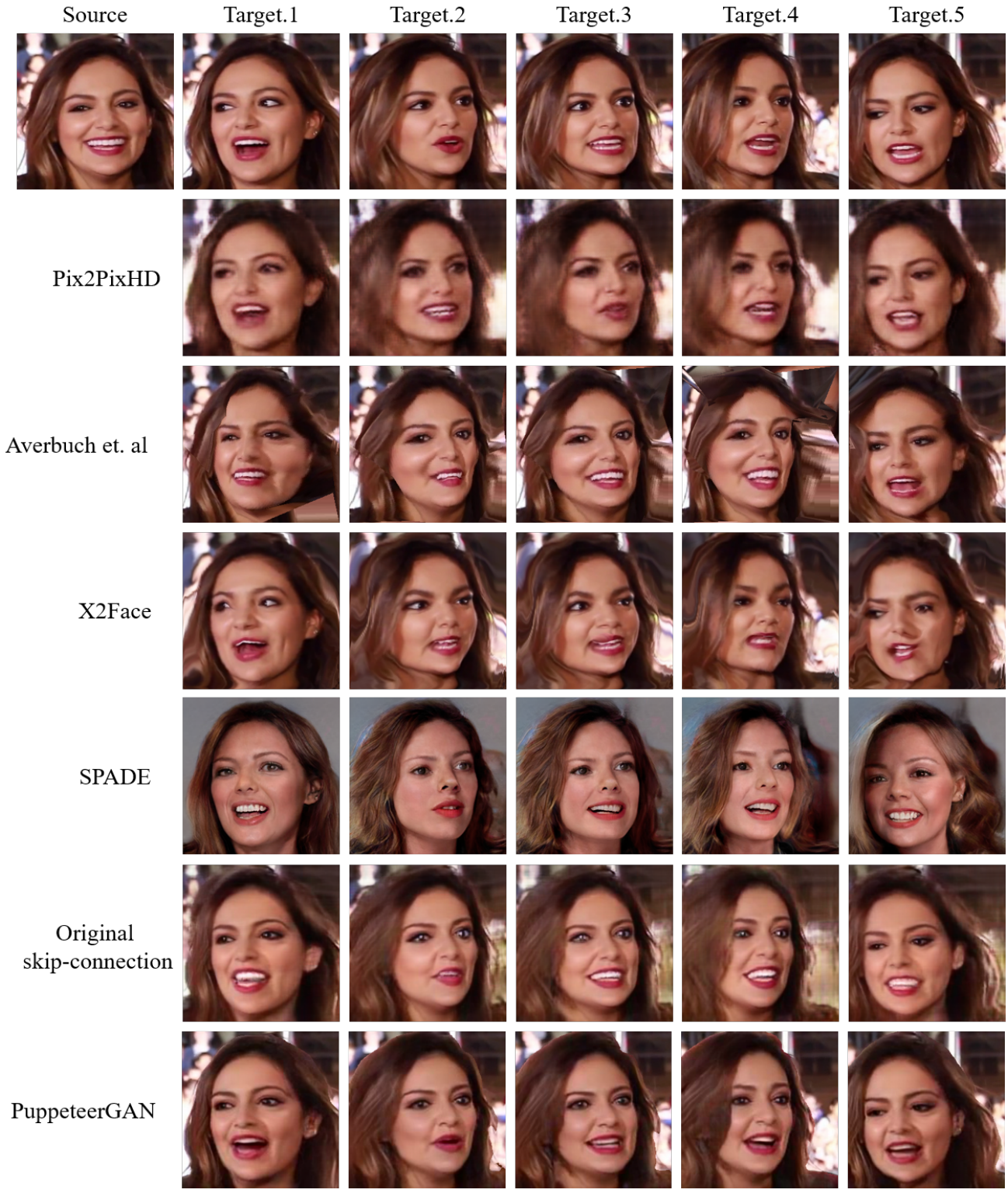

Figure 2: Another example of self-driven experiment. We show more cases with large-scale head motion in this figure compared with Fig. 1. PuppeteerGAN is able to preserve the identity of the source person and generated high realistic portraits after large scale action.

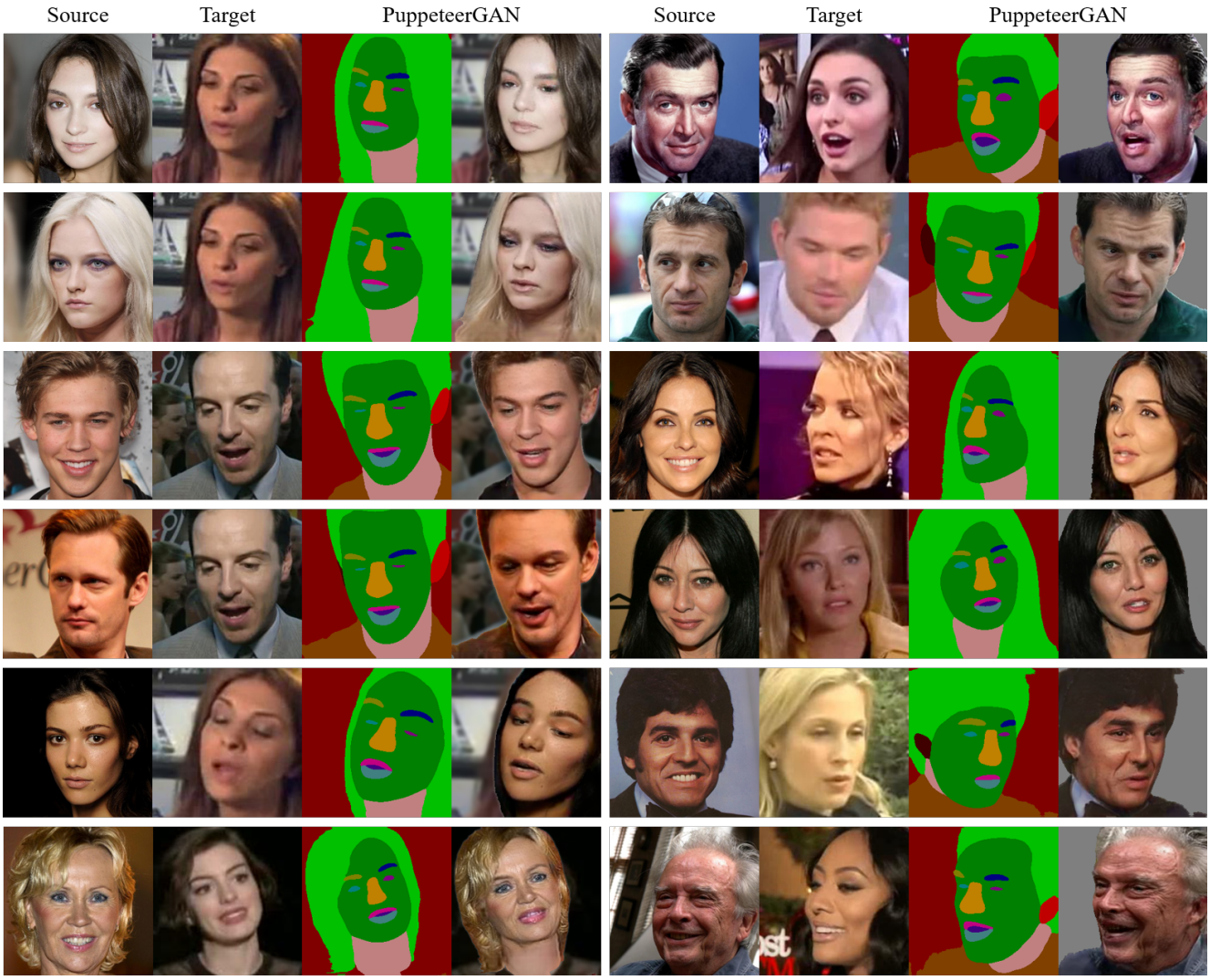

Figure 3: The results of cross-identity portrait animation experiment. For each pair, the identity of the source portrait and the target frame are different. Therefore, we perform identity preserved pose retargeting to the target frame using the Sketching Network at first. The generated segmentation mask is shown in the third place of each example. And then the Coloring Network transforms the appearance from the generated segmentation mask to a realistic portrait with the same identity as the source person. The generated portraits are shown in the last location of each example. Our results demonstrate the high quality of our method to preserve identity even with large scale pose changing.

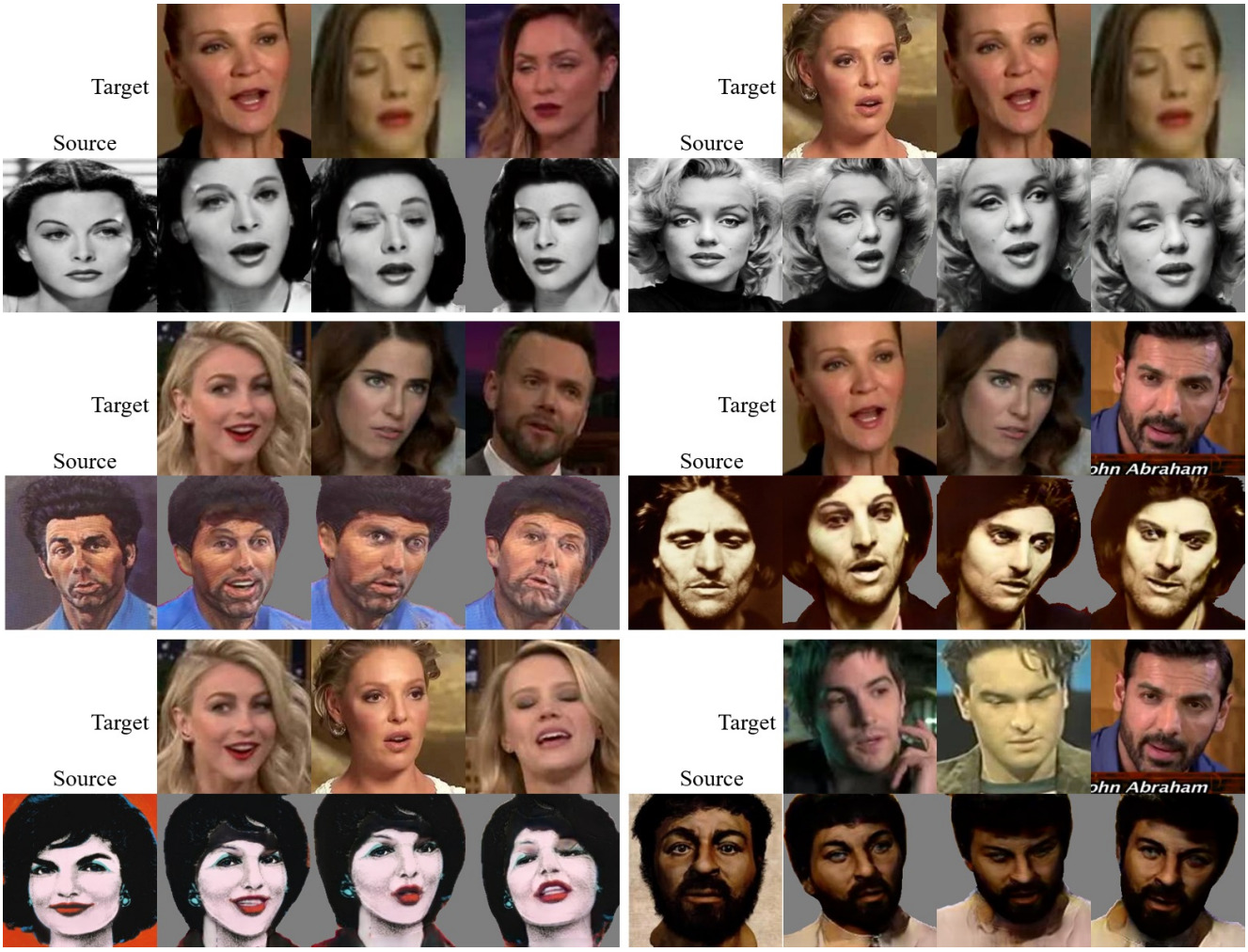

Figure 4: Cross-domain portrait animation examples. We show the results of animating black-and-white photos and different kinds of paintings. For each example, the source portrait is animated to three poses provided by different people with various pose displayed in the first row. The second line shows the source image and three generated portraits.

Figure 5: Cross-domain portrait animation examples. PuppeteerGAN is able to animate colorful photos, oil paintings and sculptures without any fine-tuning training. The generated portraits is consistent with the source image in appearance including identity and texture, while keeping at the same pose as the target person.

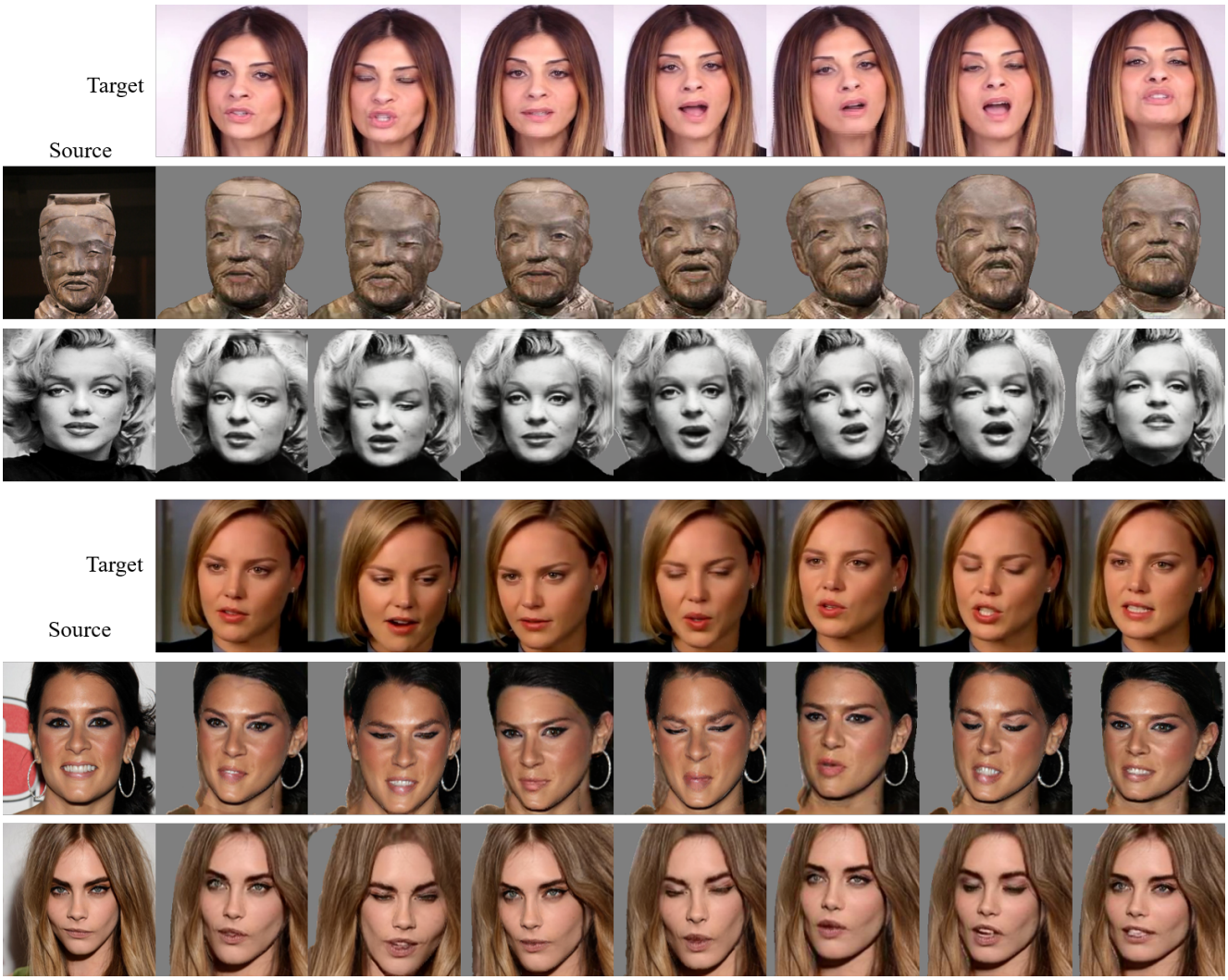

Figure 6: Sequence-driven portrait animation results. We show the results of animating two portraits of different domains or identities by one driven sequence. For each test, the first column shows the source portraits and the second row displays the target frames. The results of one source portrait animated by different target frames are continuous and consistent in identity and texture.

Figure 7: Cross-resolution portrait animation results. The first column shows the source image, and the next three columns are the generated portraits with the target frame at the right top corner. Because of the separation of pose retargeting and appearance transformation, our Coloring Network can be trained on image datasets. After being trained on CelebAMask-HQ [2] Dataset with the resolution of 512×512, PuppeteerGAN can animate high-resolution portraits without fine-tuning.

References

[1] Hadar Averbuch-Elor, Daniel Cohen-Or, Johannes Kopf, and Michael F Cohen. Bringing portraits to life. *ACM Transactions on Graphics (TOG)*, 36(6):196, 2017. 1, 4

[2] Cheng-Han Lee, Ziwei Liu, Lingyun Wu, and Ping Luo. Maskgan: Towards diverse and interactive facial image manipulation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 10

[3] Arsha Nagrani, Joon Son Chung, and Andrew Zisserman. Voxceleb: a large-scale speaker identification dataset. *Telephony*, 3:33–039, 2017. 1

[4] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2337–2346, 2019. 1, 4

[5] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8798–8807, 2018. 1, 4

[6] Olivia Wiles, A Sophia Koepke, and Andrew Zisserman. X2face: A network for controlling face generation using images, audio, and pose codes. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 670–686, 2018. 1, 4